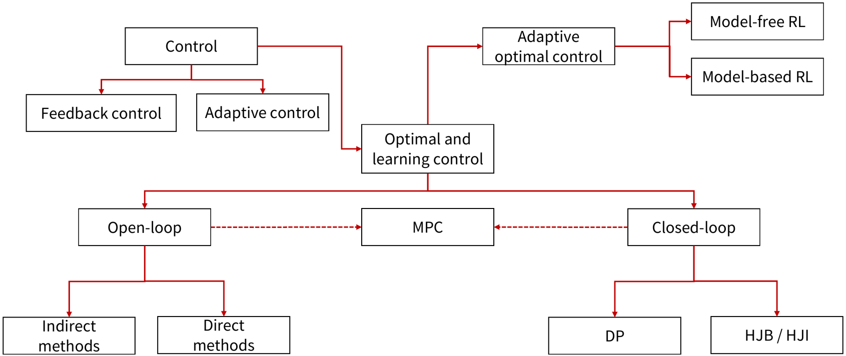
AA 203
Optimal and Learning-Based Control
Course review

Spencer M. Richards

Autonomous Systems Laboratory, Stanford University

June 7, 2023
(last updated June 12, 2023)

Course overview and goals



- To learn the *theory* and *practice* of fundamental techniques in optimal and learning-based control.
- To gain a *holistic understanding* of how such techniques are used across fields.

Agenda

1. Optimal control problems
2. Nonlinear optimization theory
3. Pontryagin's maximum principle and indirect methods
4. Direct methods
5. LQR-based methods
6. Model predictive control

Agenda

1. Optimal control problems
2. Nonlinear optimization theory
3. Pontryagin's maximum principle and indirect methods
4. Direct methods
5. LQR-based methods
6. Model predictive control

Optimal control problems (continuous-time)

$$
\begin{aligned}
\underset{x,u}{\text{minimize}} \quad & J(x,u) \coloneqq \ell_T(T, x(T)) + \int_0^T \ell(t, x(t), u(t))\, dt && \text{cost (terminal + stage)} \\
\text{subject to} \quad & \dot{x}(t) = f(t, x(t), u(t)),\ \forall t \in [0,T] && \text{dynamical feasibility} \\
& x(t_0) = x_0,\ x(T) \in \mathcal{X}_T && \text{boundary conditions} \\
& x(t) \in \mathcal{X},\ \forall t \in [0,T] && \text{state constraints} \\
& u(t) \in \mathcal{U},\ \forall t \in [0,T] && \text{input constraints}
\end{aligned}
$$

An optimal control $u^*(t)$ for a specific initial state x_0 is an *open-loop* input. An optimal control of the form $u^*(t) = \pi^*(t, x(t))$ is a *closed-loop* input.

Optimal control problems (discrete-time)

$$
\begin{aligned}
\underset{x,u}{\text{minimize}} \quad & J(x,u) := \ell_T(T, x_T) + \sum_{t=0}^{T-1} \ell(t, x_t, u_t) && \text{cost (terminal + stage)} \\
\text{subject to} \quad & x_{t+1} = f(t, x_t, u_t),\ \forall t \in \{0, 1, \ldots, T-1\} && \text{dynamical feasibility} \\
& x_0 = \bar{x}_0,\ x_T \in \mathcal{X}_T && \text{boundary conditions} \\
& x_t \in \mathcal{X},\ \forall t \in \{0, 1, \ldots, T-1\} && \text{state constraints} \\
& u_t \in \mathcal{U},\ \forall t \in \{0, 1, \ldots, T-1\} && \text{input constraints}
\end{aligned}
$$

An optimal control u_t^* for a specific initial state x_0 is an *open-loop* input. An optimal control of the form $u_t^* = \pi^*(t, x_t)$ is a *closed-loop* input.

Agenda

Unconstrained optimization

Given an objective function $f : \mathbb{R}^n \to \mathbb{R}$, we denote an *unconstrained nonlinear program* with the notation

$$\underset{x \in \mathbb{R}^n}{\text{minimize}}\, f(x).$$

We usually assume either $f \in \mathcal{C}^1$ (i.e., "continuously differentiable") or $f \in \mathcal{C}^2$ (i.e., "twice continuously differentiable").

A solution candidate $x^* \in \mathbb{R}^n$ can be a:

local minimum $\exists \varepsilon > 0 : f(x^*) \leq f(x), \ \forall x : \|x - x^*\| \leq \varepsilon$

global minimum $f(x^*) \leq f(x), \ \forall x \in \mathbb{R}^n$

If the inequality is strict, i.e., "$<$", then x^* is a strict unconstrained local/global minimum. Any (strict) global minimum is also a (strict) local minimum.

There can be many minima, or none at all!

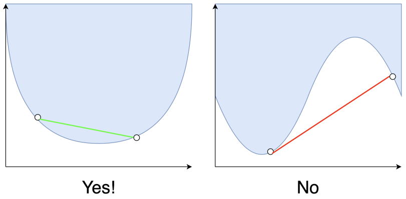
Convex sets and convex functions

A set $\mathcal{X} \subseteq \mathbb{R}^n$ is *convex* if

$$\alpha x + (1-\alpha) y \in \mathcal{X}, \ \forall x, y \in \mathcal{X}, \ \forall \alpha \in [0,1].$$

A function $f : \mathcal{X} \to \mathbb{R}^n$ is *convex* on $\mathcal{X}$ if

$$f(\alpha x + (1-\alpha) y) \leq \alpha f(x) + (1-\alpha) f(y), \quad \forall x, y \in \mathcal{X}, \ \forall \alpha \in [0,1].$$



If the inequality is strict, then f is *strictly convex*.

A function $f \in \mathcal{C}^2$ is convex on $\mathcal{X}$ if and only if $\nabla^2 f(x) \succeq 0$ for all $x \in \mathcal{X}$. If $\nabla^2 f(x) \succ 0$ for all $x \in \mathcal{X}$, then f is strictly convex.

Important examples of convex functions for this course are:

- **Quadratic** $f(x) = x^{\mathsf{T}} Q x$ (where $Q \succeq 0$)
- **Affine** $f(x) = Ax + b$ (both convex and concave)

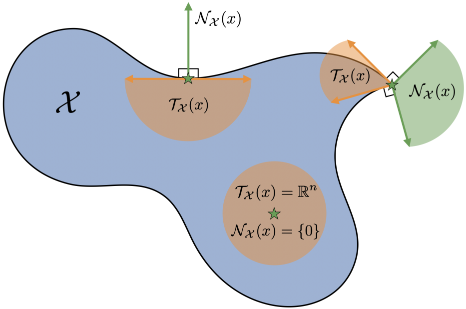
Geometry of first-order NOCs

Tangent cone $\mathcal{T}_\mathcal{X}(x)$ "vectors that stay in $\mathcal{X}$"

Normal cone $\mathcal{N}_\mathcal{X}(x)$ "vectors that leave $\mathcal{X}$"

If x^* is a local minimum of f over $\mathcal{X}$, then $-\nabla f(x^*) \in \mathcal{N}_\mathcal{X}(x^*)$, i.e., there is no feasible component of $-\nabla f(x^*)$ that would allow us to locally decrease $f(x^*)$.

For convenience, we write "$-\nabla f(x^*) \perp_{x^*} \mathcal{X}$". In other literature, you may see "$-\nabla f(x^*) \perp \mathcal{T}_\mathcal{X}(x^*)$".



If $\mathcal{X} = \{x \in \mathbb{R}^n \mid h(x) = 0,\ g(x) \preceq 0\}$ and the LICQ holds at $x^* \in \mathcal{X}$, then

$$\mathcal{T}_\mathcal{X}(x^*) = \left\{ d \in \mathbb{R}^n \mid \frac{\partial h}{\partial x}(x^*) d = 0,\ \nabla g_j(x^*)^\mathsf{T} d \leq 0,\ \forall j \in \mathcal{A}_g(x^*) \right\}$$

$$\mathcal{N}_\mathcal{X}(x^*) = \left\{ v \in \mathbb{R}^n \mid v = \frac{\partial h}{\partial x}(x^*)^\mathsf{T} \lambda + \frac{\partial g}{\partial x}(x^*)^\mathsf{T} \mu,\ \mu \succeq 0,\ \mu_j = 0, \forall j \notin \mathcal{A}_g(x^*) \right\}$$

Fritz John first-order NOCs

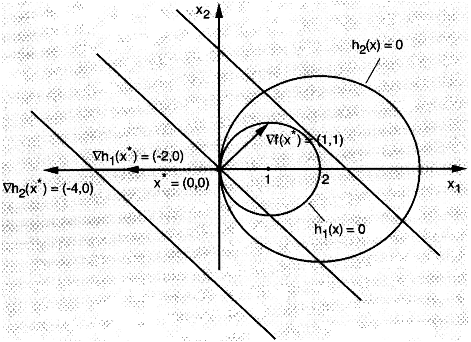
Theorem (Fritz John first-order NOCs)

If x^* is a local minimum, there exist $(\eta, \lambda^*, \mu^*) \in \{0,1\} \times \mathbb{R}^m \times \mathbb{R}^r$ such that

$$(\eta, \lambda^*, \mu^*) \neq 0$$
$$-\nabla_x L_\eta(x^*, \lambda^*, \mu^*) \perp_{x^*} \mathcal{S}$$
$$\mu_j^* \geq 0, \;\; \mu_j^* g_j(x^*) = 0, \;\; \forall j \in \{1, 2, \ldots, r\}$$

where $L_\eta(x, \lambda, \mu)$ is the partial Lagrangian

$$L_\eta(x, \lambda, \mu) := \eta f(x) + \lambda^\mathsf{T} h(x) + \mu^\mathsf{T} g(x).$$



The "abnormal case" $\eta = 0$ yields necessary conditions independent of the objective f.

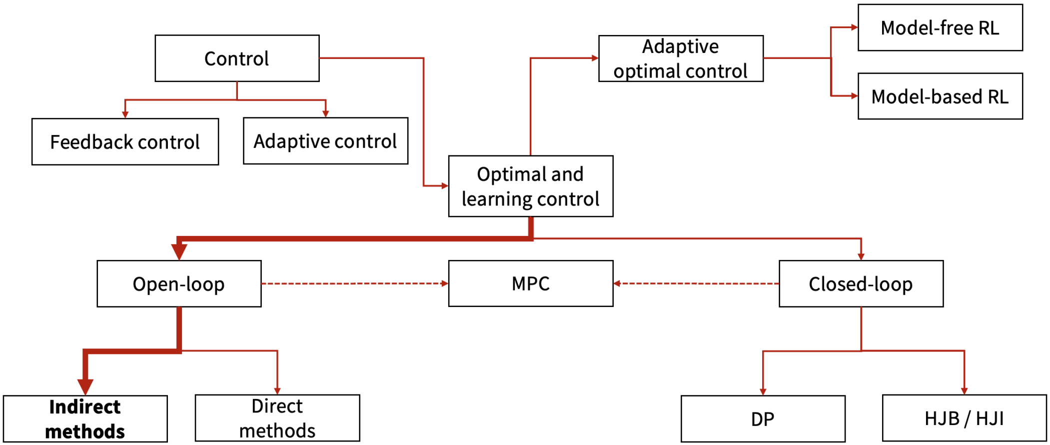
Corollary

If $\mathcal{S} = \mathbb{R}^n$ and the LICQ holds, then $\eta = 1$ and $\nabla_x L_1(x^*, \lambda^*, \mu^*) = 0$.

Agenda

1. Optimal control problems
2. Nonlinear optimization theory
3. Pontryagin's maximum principle and indirect methods
4. Direct methods
5. LQR-based methods
6. Model predictive control

Course overview



Pontryagin maximum principle (discrete-time)

Collect these necessary conditions together to get the Pontryagin maximum principle (PMP).

Theorem (Pontryagin maximum principle (discrete-time))

Let (x^*, u^*) be a local minimum of the discrete-time OCP with terminal set $\mathcal{X}_T$ and control set $\mathcal{U}$. Then $\eta \in \{0, 1\}$ and $\{p_t^*\}_{t=0}^T \subset \mathbb{R}^n$ exist such that

$$(\eta, p_0^*, p_1^*, \ldots, p_T^*) \neq 0 \qquad \textit{non-triviality}$$

$$p_t^* = \nabla_x H_\eta(t, x_t^*, u_t^*, p_{t+1}^*),\ \forall t \in \{0, 1, \ldots, T-1\} \qquad \textit{adjoint equation}$$

$$-p_T^* - \eta \nabla \ell_T(x_T^*) \perp_{x_T^*} \mathcal{X}_T \qquad \textit{transversality}$$

$$\nabla_u H_\eta(t, x_t^*, u_t^*, p_{t+1}^*) \perp_{u_t^*} \mathcal{U},\ \forall t \in \{0, 1, \ldots, T-1\} \qquad \textit{maximum condition (weak)}$$

Pontryagin maximum principle (continuous-time)

The possibility of large spatial control perturbations still corresponding to "feasible neighbours" of (x^*, u^*) suggests the following strengthened PMP.

Theorem (Pontryagin maximum principle (continuous-time))

Let (x^*, u^*) be a local minimum (using the $\mathcal{C}^0$-norm and $\mathcal{L}^1$-norm, respectively) of the continuous-time OCP with terminal set $\mathcal{X}_T$ and bounded control set $\mathcal{U}$. Then $\eta \in \{0, 1\}$ and $p^* : [0, T] \to \mathbb{R}^n$ exist such that

$$
\begin{aligned}
(\eta, p^*(t)) &\not\equiv 0 && \textit{non-triviality} \\
-\dot{p}^*(t) &= \nabla_x H_\eta(t, x^*(t), u^*(t), p^*(t)),\ \forall t \in [0, T] && \textit{adjoint equation} \\
-p^*(T) - \eta \nabla \ell_T(x^*(T)) &\perp_{x^*(T)} \mathcal{X}_T && \textit{transversality} \\
H_\eta(t, x^*(t), u^*(t), p^*(t)) &= \sup_{u \in \mathcal{U}} H_\eta(t, x^*(t), u, p^*(t)),\ \forall t \in [0, T] && \textit{maximum condition}
\end{aligned}
$$

A rigorous proof relies on variational calculus (Liberzon, 2012; Clarke, 2013).

An *indirect method* generally focuses on solving the BVP

$$\begin{pmatrix} \dot{x}^* \\ \dot{p}^* \end{pmatrix} = \begin{pmatrix} f(t, x^*, u^*) \\ -\nabla_x H_\eta(t, x^*, u^*(t, x^*, p^*), p^*) \end{pmatrix}, \quad x^*(0) = x_0, \quad h(x^*(T), p^*(T)) = 0.$$

where $h(x^*(T), p^*(T)) \in \mathbb{R}^n$. The *open-loop* optimal control candidate $u^*(t, x^*(t), p^*(t))$ is then extracted.

The boundary condition $h(x^*(T), p^*(T)) = 0$ is determined by the terminal set constraint $x^*(T) \in \mathcal{X}_T$ and the transversality condition $-p^*(T) - \eta \nabla \ell_T(x^*(T)) \perp_{x^*(T)} \mathcal{X}_T$.

We are implicitly assuming an optimal control exists. Even then, there may be multiple local optima.

Shooting methods

To solve the BVP

$$\begin{pmatrix} \dot{x}^* \\ \dot{p}^* \end{pmatrix} = \begin{pmatrix} f(t, x^*, u^*) \\ -\nabla_x H_\eta(t, x^*, u^*(t, x^*, p^*), p^*) \end{pmatrix}, \quad x^*(0) = x_0, \quad h(x^*(T), p^*(T)) = 0,$$

we consider the associated initial value problem (IVP)

$$\begin{pmatrix} \dot{x}^* \\ \dot{p}^* \end{pmatrix} = \begin{pmatrix} f(t, x^*, u^*) \\ -\nabla_x H_\eta(t, x^*, u^*(t, x^*, p^*), p^*) \end{pmatrix}, \quad x^*(0) = x_0, \quad p^*(0) = p_0.$$

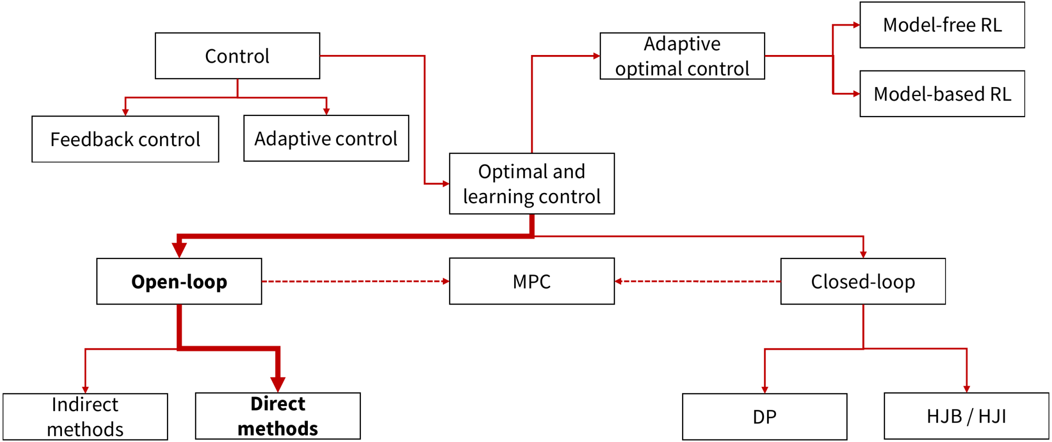
We can integrate the IVP forward in time to get $x^*(T; p_0)$ and $p^*(T; p_0)$, which are parameterized by p_0.

We can use a root-finding method (e.g., bisection search, Newton-Raphson method) to find p_0 such that $h(x^*(T; p_0), p^*(T; p_0)) = 0$. This is called *single shooting* and gives us a solution of the BVP.

Agenda

1. Optimal control problems
2. Nonlinear optimization theory
3. Pontryagin's maximum principle and indirect methods
4. Direct methods
5. LQR-based methods
6. Model predictive control

Course overview



In *direct methods*, we *transcribe* the OCP into a *finite-dimensional* nonlinear optimization problem that we then solve directly with nonlinear programming.

$$\left.\begin{aligned}
\underset{x,u}{\text{minimize}}\ \ & \ell_T(x(T)) + \int_0^T \ell(t,x(t),u(t))\,dt \\
\text{subject to}\ \ & \dot{x}(t) = f(t,x(t),u(t)),\ \forall t \in [0,T] \\
& x(0) = x_0 \\
& x(T) \in \mathcal{X}_T \\
& u(t) \in \mathcal{U},\ \forall t \in [0,T]
\end{aligned}\right\} \implies \left\{\begin{aligned}
\underset{z\in\mathcal{S}\subseteq\mathbb{R}^d}{\text{minimize}}\ \ & f(z) \\
\text{subject to}\ \ & h(z) = 0 \\
& g(z) \preceq 0
\end{aligned}\right.$$

Each direct method uses some manner of *transcription*, which also determines what z, h, g, and $\mathcal{S}$ represent.

Sequential convex programming (SCP)

Consider the non-convex problem

$$\begin{aligned} \underset{z \in \mathbb{R}^d}{\text{minimize}} \ & f(z) \\ \text{subject to} \ & h(z) = 0 \\ & g(z) \preceq 0 \end{aligned}$$

The basic idea of SCP is to iteratively solve for $z^{(k)}$ via the convex sub-problem

$$\begin{aligned} \underset{z \in \mathbb{R}^d}{\text{minimize}} \ & \hat{f}^{(k)}(z) \\ \text{subject to} \ & \hat{h}^{(k)}(z) := \hat{A}^{(k)} z - \hat{b}^{(k)} = 0 \\ & \hat{g}^{(k)}(z) \preceq 0 \\ & z \in \mathcal{T}^{(k)} := \{ z \in \mathbb{R}^d \mid \| z - z^{(k)} \|_\infty \leq \rho^{(k)} \} \end{aligned}$$

where $(\hat{f}^{(k)}, \hat{g}^{(k)})$ and $\hat{h}^{(k)}$ are convex and affine, respectively, *approximations* of (f, g) and h, respectively, over a convex *trust region* $\mathcal{T}^{(k)}$ around $z^{(k)}$ for some $\rho^{(k)} > 0$.

Consider the discrete-time OCP

$$\begin{aligned}
\underset{x,u}{\text{minimize}} \quad & \ell_T(x_T) + \sum_{t=0}^{T-1} \ell(t, x_t, u_t) \\
\text{subject to} \quad & x_{t+1} = f(t, x_t, u_t),\ \forall t \in \{0, 1, \ldots, T-1\} \\
& x_0 = \bar{x}_0 \\
& x_T \in \mathcal{X}_T \\
& u_t \in \mathcal{U},\ \forall t \in \{0, 1, \ldots, T-1\}
\end{aligned}$$

Assume $\mathcal{X}_T$ and $\mathcal{U}$ are convex sets, and ℓ_T and ℓ are convex in x_T and (x_t, u_t), respectively. Then the remaining non-convexity is due to the nonlinear dynamics constraints.

SCP for OCPs

Let $\{x_t^{(k)}\}_{t=0}^{T}$ and $\{u_t^{(k)}\}_{t=0}^{T-1}$ represent our current solution iterate. Linearize the dynamics around this iterate to get the estimate

$$\hat{f}^{(k)}(t, x_t, u_t) \coloneqq f(t, x_t^{(k)}, u_t^{(k)}) + \frac{\partial f}{\partial x}(t, x_t^{(k)}, u_t^{(k)})(x_t - x_t^{(k)}) + \frac{\partial f}{\partial u}(t, x_t^{(k)}, u_t^{(k)})(u_t - u_t^{(k)})$$

which allows us to construct the *convex* OCP

$$\begin{aligned}
\underset{x,u}{\text{minimize}} \quad & \ell_T(x_T) + \sum_{t=0}^{T-1} \ell(t, x_t, u_t) \\
\text{subject to} \quad & x_{t+1} = \hat{f}^{(k)}(t, x_t, u_t),\ \forall t \in \{0, 1, \ldots, T-1\} \\
& x_0 = \bar{x}_0 \\
& x_T \in \mathcal{X}_T \\
& u_t \in \mathcal{U},\ \forall t \in \{0, 1, \ldots, T-1\} \\
& \|x_t - x_t^{(k)}\|_\infty \leq \rho_x^{(k)},\ \forall t \in \{0, 1, \ldots, T-1\} \\
& \|u_t - u_t^{(k)}\|_\infty \leq \rho_u^{(k)},\ \forall t \in \{0, 1, \ldots, T-1\}
\end{aligned}$$

Direct methods in practice

"As you begin to play with these algorithms on your own problems, you might feel like you're on an emotional roller-coaster." - Russ Tedrake, *Underactuated Robotics*

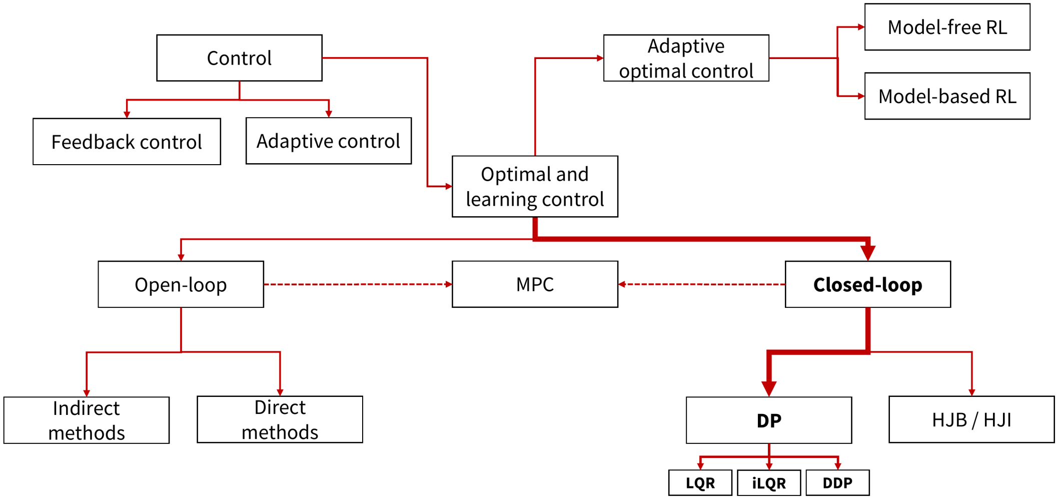
In general, there are no guarantees for solving nonlinear optimization problems. You can converge to a bad local minimum, or not at all.

You may need to spend some time tuning, e.g., your cost function and trust region radii, or perhaps adding slack variables.

It is also a good idea to try "warm-starting" your initial guess in SCP with the solution to an easier problem (e.g., one with looser constraints).

Agenda

Course overview



LQR feedback for linear systems with quadratic costs

Consider the discrete-time OCP

$$\begin{aligned}\underset{u}{\text{minimize}}\ \ & \frac{1}{2}x_T^\mathsf{T} Q_T x_T + \sum_{t=0}^{T-1}\left(\frac{1}{2}x_t^\mathsf{T} Q_t x_t + \frac{1}{2}u_t^\mathsf{T} R_t u_t + x_t^\mathsf{T} S_t u_t\right)\\ \text{subject to}\ \ & x_{t+1} = A_t x_t + B_t u_t,\ \ \forall t \in \{0, 1, \dots, T-1\}\end{aligned}$$

which is parameterized by the initial state x_0 and minimized over the control inputs u alone, for $Q_T \succeq 0$, $Q_t \succeq 0$, and $R_t \succ 0$.

We solved this recursively via dynamic programming, during which we encountered the Bellman optimality equation

$$J_t^*(x_t) = \min_{u_t} \underbrace{\frac{1}{2}\left(\begin{pmatrix} x_t \\ u_t \end{pmatrix}^\mathsf{T} \begin{bmatrix} Q_t & S_t \\ S_t^\mathsf{T} & R_t \end{bmatrix} \begin{pmatrix} x_t \\ u_t \end{pmatrix} + \underbrace{(A_t x_t + B_t u_t)^\mathsf{T} P_{t+1} (A_t x_t + B_t u_t)}_{=J_{t+1}^*(x_{t+1})}\right)}_{\text{state-action value function } Q^*(x_t, u_t)}$$

LQR feedback for linear systems with quadratic costs

Consider the discrete-time OCP

$$\begin{aligned} \underset{u}{\text{minimize}} \;\; & J_0(x_0) := \frac{1}{2}x_T^\mathsf{T} Q_T x_T + \sum_{t=0}^{T-1}\left(\frac{1}{2}x_t^\mathsf{T} Q_t x_t + \frac{1}{2}u_t^\mathsf{T} R_t u_t + x_t^\mathsf{T} S_t u_t\right) \\ \text{subject to} \;\; & x_{t+1} = A_t x_t + B_t u_t,\; \forall t \in \{0, 1, \dots, T-1\} \end{aligned}$$

which is parameterized by $x_0 \in \mathbb{R}^n$, $Q_T \succeq 0$, $Q_t \succeq 0$, and $R_t \succ 0$.

The optimal control $u^* = \pi^*(t, x) = K_t x$ is *closed-loop* and *linear*. It can be computed offline via the backwards Riccati recursion

$$\begin{aligned} P_T &:= Q_T \\ K_t &= -(R_t + B_t^\mathsf{T} P_{t+1} B_t)^{-1}(B_t^\mathsf{T} P_{t+1} A_t + S_t^\mathsf{T}) \\ P_t &= Q_t + A_t^\mathsf{T} P_{t+1} A_t - (A_t^\mathsf{T} P_{t+1} B_t + S_t)(R_t + B_t^\mathsf{T} P_{t+1} B_t)^{-1}(B_t^\mathsf{T} P_{t+1} A_t + S_t^\mathsf{T}) \\ &= Q_t + A_t^\mathsf{T} P_{t+1}(A_t + B_t K_t) + S_t K_t \end{aligned}$$

LQR-based methods for solving unconstrained nonlinear OCPs

Consider the discrete-time OCP

$$\begin{aligned}
\underset{\bar{x},\bar{u}}{\text{minimize}} \quad & J(\bar{x},\bar{u}) \coloneqq \ell_T(\bar{x}_T) + \sum_{t=0}^{T-1} \ell(t,\bar{x}_t,\bar{u}_t) \\
\text{subject to} \quad & \bar{x}_{t+1} = f(t,\bar{x}_t,\bar{u}_t), \ \forall t \in \{0,1,\dots,T-1\} \\
& \bar{x}_0 = x_0
\end{aligned}$$

We can use LQR to approximately solve this problem for an *open-loop* trajectory $(\bar{x},\bar{u})$ and a *locally optimal policy* $u_t^* = \pi_t^*(t,x_t,\bar{x}_t,\bar{u}_t) = \bar{u}_t + K_t(x_t - \bar{x}_t)$ simultaneously!

Specifically, we will consider two *iterative* methods:

iterative LQR (iLQR) Approximate the cost and dynamics as quadratic and affine, respectively, then solve the optimal Bellman equation recursively.

differential dynamic programming (DDP) Approximate the value function and Bellman equation as quadratic, then solve the optimal Bellman equation recursively.

iLQR and DDP

Input: initial state $x_0 \in \mathbb{R}^n$, convergence tolerance $\varepsilon > 0$, maximum iterations $N \in \mathbb{N}_{>0}$

initialize nominal control sequence $\bar{u} = \{\bar{u}_t\}_{t=0}^{T-1}$, initial cost change $\tilde{J} = \infty$.

Rollout $\bar{x}_{t+1} = f(t, \bar{x}_t, \bar{u}_t)$ to get $\bar{x} = \{\bar{x}_t\}_{t=0}^{T}$ and $J(\bar{x}, \bar{u})$.

for $i = 1, 2, \ldots, N$

Backward pass:

Compute the approximating terms $\{\eta_t, h_{x,t}, h_{u,t}, H_{xx,t}, H_{uu,t}, H_{xu,t}\}_{t=0}^{T-1}$.
Recursively compute $\{\beta_t, p_t, P_t\}_{t=0}^{T}$ and $\{k_t, K_t\}_{t=0}^{T-1}$.

Forward pass:

Rollout $\tilde{x}_{t+1} = f(t, \bar{x}_t + \tilde{x}_t, \bar{u}_t + \tilde{u}_t) - \bar{x}_{t+1}$ with $\tilde{u}_t = k_t + K_t \tilde{x}_t$.
Update $(\bar{x}, \bar{u}) \leftarrow (\bar{x} + \tilde{x}, \bar{u} + \tilde{u})$ and $\tilde{J} \leftarrow J(\bar{x} + \tilde{x}, \bar{u} + \tilde{u}) - J(\bar{x}, \bar{u})$.

if $\|\tilde{u}\|_\infty < \varepsilon$ and/or $|\tilde{J}| < \varepsilon$

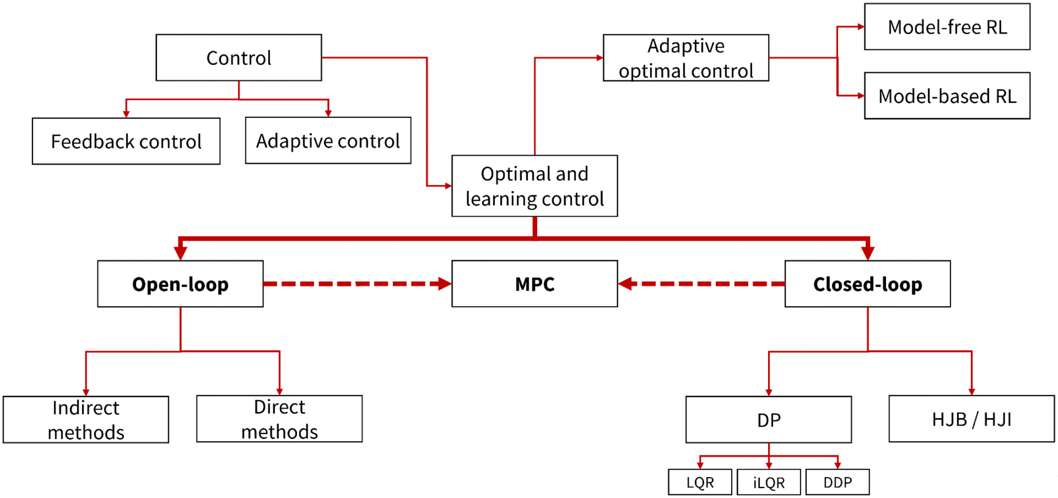
break

return $\bar{x}$, $\bar{u}$, and $\{k_t, K_t\}_{t=0}^{T-1}$.

The output is an open-loop trajectory $(\bar{x}, \bar{u})$ that is locally optimal for the OCP, and a policy $\pi(t, x, \bar{x}, \bar{u}) = \bar{u} + k_t + K_t(x - \bar{x})$ that is locally optimal for closed-loop tracking.

Agenda

Course overview



Model predictive control

Consider the discrete-time dynamical system $x(t+1) = f(x(t), u(t))$. In *model predictive control (MPC)*, at each time t we solve the optimal control problem

$$
\begin{aligned}
\underset{u}{\text{minimize}} \;\; & V_{\mathsf{f}}(x_{N|t}) + \sum_{k=0}^{N-1} \ell(x_{k|t}, u_{k|t}) \\
\text{subject to} \;\; & x_{0|t} = x(t) \\
& x_{k+1|t} = f(x_{k|t}, u_{k|t}), \quad && \forall k \in \{0, 1, \ldots, N-1\} \\
& x_{k|t} \in \mathcal{X}, && \forall k \in \{0, 1, \ldots, N-1\} \\
& x_{N|t} \in \mathcal{X}_{\mathsf{f}} \\
& u_{k|t} \in \mathcal{U}, && \forall k \in \{0, 1, \ldots, N-1\}
\end{aligned}
$$

for a sequence $\{u^*_{k|t}\}_{k=0}^{N-1}$. The *MPC feedback policy* is then $u(t) = \pi_{\mathrm{MPC}}(x(t)) \coloneqq u^*_{0|t}$.

The set of feasible initial states is

$$
\mathcal{X}_0 \coloneqq \{x \in \mathbb{R}^n \mid \exists \{u_k\}_{k=0}^{N-1} \subseteq \mathcal{U} : \{x_k\}_{k=0}^{N-1} \in \mathcal{X},\; x_N \in \mathcal{X}_{\mathsf{f}}\},
$$

where $x_0 = x$ and $x_{k+1} \coloneqq f(x_k, u_k)$ for all $k \in \{0, 1, \ldots, N-1\}$.

Recursive feasibility and stability

Suppose:

- The initial MPC problem is feasible, i.e., $x(0) \in \mathcal{X}_0$.
- There exists a feedback policy $\pi_\mathsf{f} : \mathbb{R}^n \to \mathbb{R}^m$ that renders $\mathcal{X}_\mathsf{f}$ invariant subject to state and input constraints, i.e.,

  $$\mathcal{X}_\mathsf{f} \subseteq \mathcal{X}, \qquad \pi_\mathsf{f}(x) \in \mathcal{U}, \qquad f(x, \pi_\mathsf{f}(x)) \in \mathcal{X}_\mathsf{f},$$

  for all $x \in \mathcal{X}_\mathsf{f}$.
- The function f is continuous, and ℓ and V_f are uniformly continuous. Moreover, $f(0,0) = 0$, $\pi_\mathsf{f}(0) = 0$, and ℓ and V_f are positive-definite.
- The sets $\mathcal{X}$, $\mathcal{U}$, and $\mathcal{X}_\mathsf{f}$ are closed and bounded, and each contain the origin.
- The terminal cost function $V_\mathsf{f} : \mathbb{R}^n \to \mathbb{R}$ satisfies

  $$V_\mathsf{f}(f(x, \pi_\mathsf{f}(x))) - V_\mathsf{f}(x) \leq -\ell(x, \pi_\mathsf{f}(x))$$

  for all $x \in \mathcal{X}_\mathsf{f}$.

Then the MPC feedback policy is *recursively feasible*, and *asymptotically stabilizing* with region of attraction $\mathcal{X}_0$. Much of this depends on the *terminal ingredients* $(\mathcal{X}_\mathsf{f}, \pi_\mathsf{f}, V_\mathsf{f})$.

Predictive control versus the real-world

Any type of predictive control assumes the system evolves in a predictable fashion, e.g.,

$$x(t+1) = f(x(t), u(t)),$$

with known $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$.

In reality, we often instead have

$$x(t+1) = f(x(t), u(t), w(t)),$$

with a possibly time-varying disturbance $w(t)$ that enters our system with currently unknown structure. The disturbance $w(t)$ can be structured itself, random noise, or some combination thereof.

What can we hope to do in this situation?

Goals of robust constrained control

Consider the uncertain system

$$x(t+1) = f(x(t), u(t), w(t)),$$

where we assume $w(t) \in \mathcal{W}$ for all t and some set $\mathcal{W}$.

The goal of *robust constrained control* is to design a feedback policy $u(t) = \pi(x(t))$ such that:

- The state and input robustly satisfy constraints, e.g., $x(t) \in \mathcal{X}$ and $u(t) \in \mathcal{U}$ for all t and all possible realizations of $w(t)$.
- The system is robustly stable, e.g., $x(t)$ converges to some bounded neighbourhood of the origin.
- Closed-loop trajectories are "optimal" with respect to some notion of performance, e.g., in expectation or in the worst-case.
- The set of robustly feasible initial states $\mathcal{X}_0$ is as large as possible.

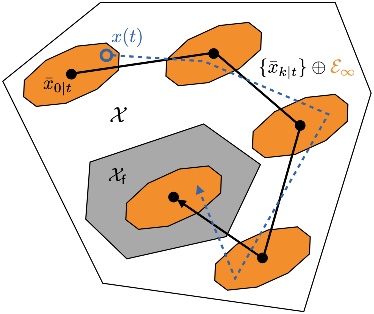
Achieving these goals requires some knowledge or assumptions about f and $\mathcal{W}$.

Tube MPC

Tube MPC computes a nominal pair $\bar{x}_{\cdot|t} = \{\bar{x}_{k|t}\}_{k=0}^{N}$ and $\bar{u}_{\cdot|t} = \{\bar{u}_{k|t}\}_{k=0}^{N-1}$, while *planning* to apply the policy

$$u_{k|t} = \bar{u}_{k|t} + K(x_{k|t} - \bar{x}_{k|t})$$

to account for future information gain.

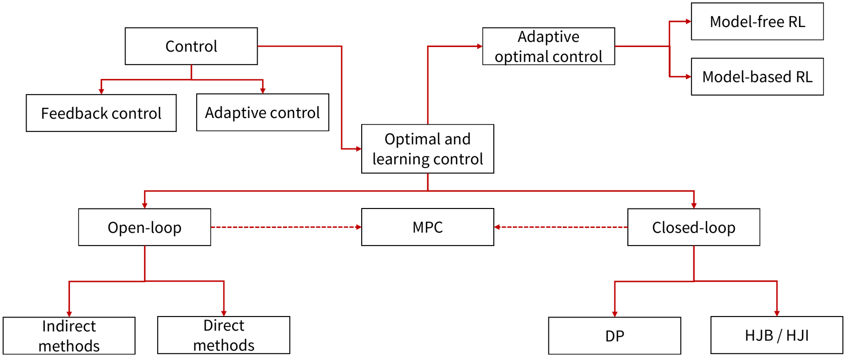


Overall, we need to:

- Compute the set $\mathcal{E}_\infty$ that the error will remain inside.
- Modify the constraints so $\{\bar{x}_{k|t}\} \oplus \mathcal{E}_\infty \subseteq \mathcal{X}$ and $\{\bar{u}_{k|t}\} \oplus K\mathcal{E}_\infty \subseteq \mathcal{U}$.
- Formulate the tube MPC problem as a convex optimization.

We can then prove that the constraints are robustly satisfied, the tube MPC problem is recursively feasible, and the closed-loop system is robustly stable.

Course overview and goals



- To learn the *theory* and *practice* of fundamental techniques in optimal and learning-based control.
- To gain a *holistic understanding* of how such techniques are used across fields.

References

F. Clarke. *Functional Analysis, Calculus of Variations and Optimal Control*. Springer, 2013.

D. Liberzon. *Calculus of Variations and Optimal Control Theory: A Concise Introduction*. Princeton University Press, 2012.